

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

(19)



JAPANESE PATENT OFFICE

JPA2003-099170

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **2003099170 A**

(43) Date of publication of application: **04.04.03**

(51) Int. Cl.

G06F 3/00
G06F 3/12

(21) Application number: **2001285681**

(71) Applicant: **CANON INC**

(22) Date of filing: **19.09.01**

(72) Inventor: **MINAGAWA TOMONORI**

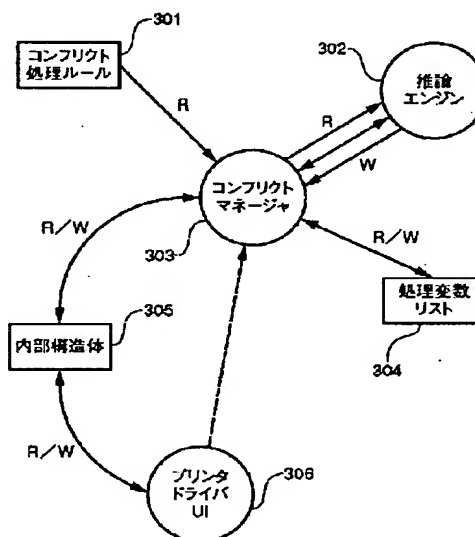
(54) **USER INTERFACE CONTROLLER, USER
INTERFACE CONTROL METHOD, COMPUTER
PROGRAM, AND STORAGE MEDIUM**

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a user interface capable of easily constructing a user interface in a window to be displayed for deciding a parameter for executing prescribed processing, and increasing productivity, and improving maintenance.

SOLUTION: When an environment setting UI is displayed at printing, an input event by an operator is monitored, and the state variables of each control are changed by referring to a conflict processing rule based on the event. Then, display change is operated so that any unavailable control can be turned to be gray out.

COPYRIGHT: (C)2003,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2003-99170

(P2003-99170A)

(43) 公開日 平成15年4月4日 (2003.4.4)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
G 0 6 F 3/00	6 5 4	G 0 6 F 3/00	6 5 4 A 5 B 0 2 1
3/12		3/12	C 5 E 5 0 1

審査請求 未請求 請求項の数10 O L (全 9 頁)

(21) 出願番号 特願2001-285681 (P2001-285681)

(22) 出願日 平成13年9月19日 (2001.9.19)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 皆川 智徳

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(74) 代理人 100076428

弁理士 大塚 康徳 (外3名)

Fターム(参考) 5B021 AA01 BB01 CC07

5E501 AA01 AA06 AC37 BA05 CA02

CB02 CB09 DA13 EA05 EA10

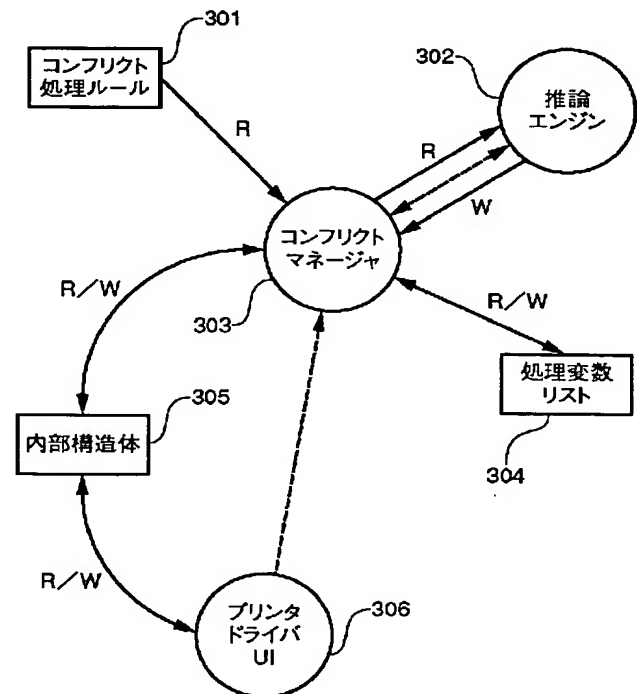
EB05 FA23 FA46 FB43

(54) 【発明の名称】 ユーザインタフェース制御装置及び方法及びコンピュータプログラム及び記憶媒体

(57) 【要約】

【課題】 所定の処理を行うためのパラメータを決定するために表示されるウインドウにおけるユーザーインタフェースを簡単に構築でき、生産性を高めると共にメンテナンス性にも優れたユーザーインタフェースを提供する。

【解決手段】 印刷する際の環境設定UIを表示させたとき、操作者による入力イベントを監視し、そのイベントに基づいてコンフリクト処理ルールを参照して、各コントロールの状態変数を変更する。そして、利用不可となるコントロールについてはグレイアウトするよう表示更新する。



【特許請求の範囲】

【請求項1】 所定の処理を行うためのパラメータを決定するため、当該複数のパラメータに対応する複数のコントロールが配置された表示ウインドウを表示し、操作者によるコントロールの操作結果に基づいて前記パラメータを決定するユーザーインタフェース制御装置であって、

前記複数のコントロールの依存関係をコンフリクト処理ルールとして管理する管理手段と、

前記複数のコントロールのうちの1つに対する操作が行われたとき、前記管理手段で管理されたコンフリクトルールに従って、各コントロールの操作可・不可を示す状態を設定する設定手段と、

該設定手段の設定結果に応じて、表示ウインドウ中のコントロールの表示形態を更新する表示更新手段とを備えることを特徴とするユーザーインタフェース制御装置。

【請求項2】 前記設定手段は、操作者の入力によるイベントをトリガにして動作することを特徴とする請求項第1項に記載のユーザーインタフェース制御装置。

【請求項3】 前記コンフリクト処理ルールは、対象コントロールの状態と、当該対象コントロールの状態を変更する原因となる他のコントロールの状態の因果関係を記述した形式となっていることを特徴とする請求項第1項に記載のユーザーインタフェース制御装置。

【請求項4】 前記対象コントロールの状態を変更する原因となる他のコントロールは複数存在することを特徴とする請求項第1項に記載のユーザーインタフェース制御装置。

【請求項5】 表示更新手段は、操作不可となるコントロールについてグレイアウトすることを特徴とする請求項第1項に記載のユーザーインタフェース制御装置。

【請求項6】 更に、対象コントロールの状態及び当該対象コントロールの状態を変更する原因となる他のコントロールの状態の因果関係と、前記対象コントロールに対するメッセージを記憶する記憶手段と、操作不可となったコントロールが指定された場合、当該指定コントロールに対応するメッセージを前記記憶手段より抽出し、表示する表示手段とを備えることを特徴とする請求項第1項に記載のユーザーインタフェース制御装置。

【請求項7】 前記表示ウインドウは、プリンタドライバの一部による印刷環境設定のウインドウとすることを特徴とする請求項第1項乃至第6項のいずれか1つに記載のユーザーインタフェース制御装置。

【請求項8】 所定の処理を行うためのパラメータを決定するため、当該複数のパラメータに対応する複数のコントロールが配置された表示ウインドウを表示し、操作者によるコントロールの操作結果に基づいて前記パラメータを決定するユーザーインタフェース制御方法であって、

前記複数のコントロールの依存関係をコンフリクト処理ルールとして管理する管理工程と、

前記複数のコントロールのうちの1つに対する操作が行われたとき、前記管理手段で管理されたコンフリクトルールに従って、各コントロールの操作可・不可を示す状態を設定する設定工程と、

該設定工程の設定結果に応じて、表示ウインドウ中のコントロールの表示形態を更新する表示更新工程とを備えることを特徴とするユーザーインタフェース制御方法。

10 【請求項9】 コンピュータが読み込み実行することで、所定の処理を行うためのパラメータを決定するため、当該複数のパラメータに対応する複数のコントロールが配置された表示ウインドウを表示し、操作者によるコントロールの操作結果に基づいて前記パラメータを決定するユーザーインタフェース制御装置として機能するコンピュータプログラムであって、

前記複数のコントロールの依存関係をコンフリクト処理ルールとして管理する管理工程のプログラムコードと、前記複数のコントロールのうちの1つに対する操作が行われたとき、前記管理手段で管理されたコンフリクトルールに従って、各コントロールの操作可・不可を示す状態を設定する設定工程のプログラムコードと、該設定工程の設定結果に応じて、表示ウインドウ中のコントロールの表示形態を更新する表示更新工程のプログラムコードとを備えることを特徴とするコンピュータプログラム。

【請求項10】 請求項第9項に記載のコンピュータプログラムを格納することを特徴とする記憶媒体。

【発明の詳細な説明】

30 【0001】

【発明の属する技術分野】本発明はユーザーインタフェース制御装置及び方法及びコンピュータプログラム及び記憶媒体に関するものである。

【0002】

【従来の技術】コンピュータ上で動作するプログラムが、ユーザーインタフェースを介してユーザからの複数の設定値の入力を受け付け、それらの設定値に基づいて何らかの処理を行う機能を有している場合、前述した複数の設定値間で、排他的な関係、依存関係等が存在する場合、ユーザからの設定値の入力を受け付ける度に、入力された値とその他の設定値の中で、関係のある設定値の値との関係を評価し、それらの設定値間に不整合がないかどうかの判別を行い、不整合があった場合にはその解消を実施している。

【0003】例えば、汎用情報処理装置で動作しているアプリケーションから、印刷指示するとき、印刷に関する各種設定を行う場合について考えてみる。

40 【0004】昨今の印刷装置の印字品位は格段に向上し、その最適化のために数多くの設定パラメータを有する。必然、GUIにおける限られたサイズの1つのウイ

ンドウでは対処できず、更に詳細については別ウインドウを表示するか、或いは、タブ形式にして表示を切り替える等で対処せざるを得ない。

【0005】しかし、設定するパラメータ（コントロール）の中には、互いに因果関係があるものも存在する。これら因果関係とは、依存関係や排他関係にあるものであるが、例えば、或る設定ウインドウで或るパラメータに関するコントロール（ラジオボタンや、オプション選択ボタン、コンボボックス等）で選択や指定を行った際、別ウインドウの依存関係のあるコントロールだけが選択や指定可能になり、排他関係にあるものは操作不可或いは利用不可（以下、コンフリクトという）とするためにグレイアウトさせる必要がある。

【0006】

【発明が解決しようとする課題】しかしながら、タブ切り替えや別ウインドウ表示を行うと、そのコンフリクト発生理由がわかりにくいという問題がある。この問題は、その理由であるメッセージを表示することで対処できるが、その為には、各コントロールに対してその操作を許可するか不許可にするかの処理も必要になるのは勿論、コントロール間のコンフリクトの発生の有無のチェックし、その理由を出力する処理も必要になる。

【0007】コンフリクトチェックを正確に行うためにはグレイアウトするための処理の各々に対応する検知処理コードが必要であり、プログラムコードの作成に手間がかかるだけでなく、グレイアウト処理の追加／変更に伴って検知処理の漏れや食い違いが出やすいという問題が発生する。従って、例えば或るコントロールを追加する場合には、そのコントロールだけでなく、他のコントロールについての因果関係を見直す必要があり、プログラム作成者の負担は増大するばかりである。本発明はかかる問題点を鑑みなされたものであり、所定の処理を行うためのパラメータを決定するために表示されるウインドウにおけるユーザーインタフェースを簡単に構築でき、生産性を高めると共にメンテナンス性にも優れたユーザーインタフェース制御装置及び方法及びコンピュータプログラム及び記憶媒体を提供しようとするものである。

【0008】

【課題を解決するための手段】この課題を解決するため、例えば本発明のユーザーインタフェース制御装置は以下の構成を備える。すなわち、所定の処理を行うためのパラメータを決定するため、当該複数のパラメータに対応する複数のコントロールが配置された表示ウインドウを表示し、操作者によるコントロールの操作結果に基づいて前記パラメータを決定するユーザーインタフェース制御装置であって、前記複数のコントロールの依存関係をコンフリクト処理ルールとして管理する管理手段と、前記複数のコントロールのうちの1つに対する操作が行われたとき、前記管理手段で管理されたコンフリクトル

ールに従って、各コントロールの操作可・不可を示す状態を設定する設定手段と、該設定手段の設定結果に応じて、表示ウインドウ中のコントロールの表示形態を更新する表示更新手段とを備える。

【0009】

【発明の実施の形態】以下、添付図面を参照して本発明に係る実施形態を詳細に説明する。

【0010】なお、実施形態では、ホストコンピュータ上で動作するアプリケーションから印刷指示する際に、印刷に関する各種設定を行う場合について説明する。なお、ホストコンピュータ上で動作するOSとしては、米国マイクロソフト社が提供するWindowsを例にするものとするが、他のOSでも同様である。

【0011】図1は本発明の実施形態を示す印刷処理システムの構成を説明するブロック図である。なお、特に断らない限り、本発明の機能が実行されるのであれば、単体の機器であっても、複数の機器からなるシステムであっても、LAN、WAN等のネットワークを介して接続が為され処理が行われるシステムであっても本発明を適用できることは言うまでもない。

【0012】図1において、3000はホストコンピュータで、1は装置全体の制御を司るCPUである。2はCPU1のワークエリアとして使用されるRAM（後述するように、OSやアプリケーション、プリンタドライバプログラム及びそのGUI設定プログラムがロードされる）であり、3はフォントROM、BIOS等を記憶するプログラム用ROM等で構成されるROMを始め、以下に説明する各要素がバス4を介して接続されている。

【0013】5はキーボードコントローラ（KBC）で、キーボード9や不図示のポインティングデバイスからのキー入力を制御する。6はCRTコントローラ（CRTC）で、CRTディスプレイ（CRT）10の表示を制御する。7はディスクコントローラ（DKC）で、ブートプログラム、各種のアプリケーション、フォントデータ、ユーザファイル、編集ファイル、プリンタ制御コマンド生成プログラム（以下プリンタドライバ）等を記憶するハードディスク（HD）、フロッピー（登録商標）ディスク（FD）等の外部メモリ11とのアクセスを制御する。外部メモリ（特にHDD）には、OSを始めアプリケーション及びプリンタドライバが格納されていて、電源投入時に、OSがRAM2にロードされ、その後ユーザーが指定することで図形編集や文書編集に関するアプリケーションがロードされ実行されることになる。

【0014】8はプリンタコントローラ（PRTC）で、所定の双方向性インターフェース（インターフェース）21を介してプリンタ1500に接続されて、プリンタ1500との通信制御処理を実行する。なお、CPU1は、例えばRAM2上に設定された表示情報RAM

へのアウトラインフォントの展開（ラスタイズ）処理を実行し、CRT10上でのWYSIWYGを可能としている。また、CPU1は、CRT10上の不図示のマウスカーソル等で指示されたコマンドに基づいて登録された種々のウィンドウを開き、種々のデータ処理を実行する。ユーザは印刷を実行する際、印刷の設定に関するウィンドウを開き、プリンタの設定や、印刷モードの選択を含むプリンタドライバに対する印刷処理方法の設定を行える。

【0015】プリンタ1500において、12はプリンタCPUで、ROM13のプログラム用ROMに記憶された制御プログラム等あるいは外部メモリ14に記憶された制御プログラム等に基づいてシステムバス15に接続される印刷部（プリンタエンジン）17に出力情報としての画像信号を出力する。また、このROM13のプログラムROMには、CPU12の制御プログラム等を記憶する。ROM13のフォント用ROMには上記出力情報を生成する際に使用するフォントデータ等を記憶し、ROM13のデータ用ROMにはハードディスク等の外部メモリ14がないプリンタの場合には、ホストコンピュータ上で利用される情報等を記憶している。CPU12は入力部18を介してホストコンピュータとの通信処理が可能となっており、プリンタ内の情報等をホストコンピュータ3000に通知可能に構成されている。19はCPU12の主メモリ、ワークエリア等として機能するRAMで、図示しない増設ポートに接続されるオプションRAMによりメモリ容量を拡張することができるように構成されている。なお、RAM19は、出力情報展開領域、環境データ格納領域、NVRAM等に用いられる。前述したハードディスク（HD）、ICカード等の外部メモリ14は、メモリコントローラ（MC）20によりアクセスを制御される。外部メモリ14は、オプションとして接続され、フォントデータ、エミュレーションプログラム、フォームデータ等を記憶する。また、18は前述した操作パネルで操作のためのスイッチおよびLED表示器等が配されている。また、前述した外部メモリは1個に限らず、少なくとも1個以上備え、内蔵フォントに加えてオプションフォントカード、言語系の異なるプリンタ制御言語を解釈するプログラムを格納した外部メモリを複数接続できるように構成されていてもよい。さらに、図示しないNVRAMを有し、操作パネル1501からのプリンタモード設定情報を記憶保持するようにしてもよい。

【0016】図2は本実施形態におけるユーザインタフェース制御プログラムがホストコンピュータ3000上のRAM2にロードされ実行可能となった状態のメモリマップを示している。なお、本実施形態におけるユーザインタフェース制御プログラムは印刷処理関連プログラム204（プリンタドライバ）の一部として存在している。図3は本実施形態のユーザインタフェース制御部の

内部処理の内容を示すシステム概要図となっている。また、図4は本実施形態のコンフリクト処理における各モジュールで扱うデータの関連を示す例を示している。図5は本実施形態におけるコンフリクト処理ルールを使ったコンフリクト処理及びグレイアウト項目の理由表示処理を示すフローチャート、図6は本実施形態におけるコンフリクト処理ルールの例を示している。そして、図7は本実施形態におけるプリンタドライバの表示するユーザインタフェースの例であり、図8は図7の状態ユーザが[Booklet Printing]を選択した直後の状態を示した例を、図9は図8の状態ユーザの操作により理由表示のイベントが発生した状態を示している。また、図11はメッセージルールの内容を示す図である。

【0017】以下、図5に示すフローチャートを中心にして本実施形態における動作を詳細に説明する。

【0018】本実施形態では、先に説明したように、プリンタドライバがCRT10に各種設定用に表示するユーザインタフェース上でのコンフリクト処理を例に説明する。

【0019】ユーザが図1に示したキーボードコントローラKBC5等により図7に示すプリンタドライバユーザインタフェース（UI）を開く指示をすることで始まる。ユーザよりプリンタドライバUIを開く指示により、OS205の管理の基、RAM2に印刷処理関連プログラム204がロードされる。印刷処理関連プログラム204がRAM2にロードされると、ユーザのプリンタドライバUIを開く指示により、まずプリンタドライバUIを開くための初期化処理が行われる。図3に示す推論エンジン302は図6に例示してあるように表記されたコンフリクト処理ルール301をコンフリクトマネージャ303を介して、RAM2中に読み込む（ステップ501）。

【0020】続いて、図4の402に示すようなコンフリクトマネージャが利用する状態変数リストを作成する（ステップ503）。コンフリクト処理ルール403の中に出現するプリンタ機能名A、B、Cのそれぞれについて同名の状態変数が存在する。この状態変数の値はプリンタドライバUIで使用される内部構造体の対応するメンバの値と連動している。全てのプリンタ機能名の状態変数の初期値はその内部構造体のメンバの値が初期値となる。

【0021】例えば、プリンタ機能名A、B、Cに対応するプリンタドライバUI内部構造体401のメンバをそれぞれcA、cB、cCとする。int cAの初期値は0なので、それに対応する状態変数Aの値はOFFとなる。従って推論エンジン302内のプリンタ機能名Aの状態値の初期値もOFFとなる。同様にプリンタ機能名Bの初期値はON、プリンタ機能名Cの初期値はOFFとなる。

50 A OFF

B ON
C OFF

次に、コンフリクト処理ルール403を発動し、記載されている

A(ON) ← B(ON), C(OFF).

が成立した場合、すなわちプリンタ機能名BがONで且つプリンタ機能名CがOFFの場合、推論エンジン302は左辺のプリンタ機能名Aの状態変数値をONに変更する。

A ON
B ON
C OFF

このように状態変数リストが初期化される。

【0022】コンフリクトチェックの推論が終了した後、コンフリクトマネージャは変更された状態変数の値をプリンタドライバUI内部構造体401の対応するメンバint cAに反映させる。つまりint cAは上記ルールが成立した事によって0から1に変更される。

【0023】続いて、プリンタドライバUIのオープンのために必要な初期化処理を行い、図7に例示するようなプリンタドライバUIをオープンする(ステップ504)。

【0024】プリンタドライバUIがオープンされた後は、OSより送られてくるイベントの取得とその処理を繰り返す(ステップ505)。なお、ここで言う「イベント」とは、現在動作しているプリンタドライバUIに対してユーザーが何らかの操作を行ったとき、その「操作」があったことOSが通知するものである。例えば、或るボタン(コントロールの1つ)アイコンをクリックした場合には、そのボタンに対してクリックされたというイベントが発生するので、そのイベントが発生した際の動作を記述しておくことで、そのイベントに対応する処理が実現できる。

【0025】ステップ505にて取得したイベントがユーザーがプリンタドライバUI上の設定項目を変更したイベントであるかどうかの判別を行い(ステップ506)、そうであった場合には、ステップ501で構築したコンフリクト処理ルールを適用する(ステップ507)。ユーザの設定変更要求が図7に示すPrint Styleを1-Sided PrintingからBooklet Printingに変更するものであった場合を例にとると、コンフリクト処理の適用は図6に示すコンフリクト処理ルールに対して行われる。

【0026】プリンタドライバUI内部構造体401のメンバとして存在するCollate、Group、Staple、Layoutの各メンバのコンフリクト処理ルール適用前の値は、以下の通りとなる。

Collate OFF
Group ON
Staple OFF

Layout 1-SIDED

ユーザの変更要求がLayoutを1-SidedからBookletに変更するものであるので、Layoutのメンバの内容は下記の通りとなる。

Collate OFF
Group ON
Staple OFF
Layout BOOKLET

すると、プリンタドライバUI306はコンフリクトマネージャ303を呼び出し、状態変数リストにあるLayoutの状態変数が更新され、続いて推論エンジン302がコールされて、コンフリクト処理ルールの適用が始まる。まず推論エンジン302内の各プリンタ機能名が状態変数リストの各メンバの持つ値で初期化される。続いて、図6の(5)が適用され、StapleはOFFのままコントロールはdisable、すなわちグレイアウトされる。同様に図6(7)、(8)が適用され、Collateの値はOFFからONへ、Groupの値はONからOFFへ変更され、各コントロールはdisableとなる。

20 Collate ON (disable)
Group OFF (disable)
Staple OFF (disable)
Layout BOOKLET

以上で推論エンジン302でのコンフリクト処理ルールの適用が終了する。

【0027】次のステップでは、コンフリクトマネージャ303が上記の最終状態を元に状態変数リストの更新(ステップ508)とプリンタドライバUI内部構造体の更新(ステップ509)を行う。

30 【0028】続いて、プリンタドライバUI306がプリンタドライバUI内部構造体のメンバの値を参照して、UIの更新が必要かどうかの判別を行う(ステップ510)。UIの更新の必要がない場合には、そのままステップ505に戻る。更新が必要な場合にはUIの更新処理を行う(ステップ511)。またこのステップでは、後述する理由表示のために任意のコントロールにコンフリクトマークをつける処理も行う。上記の例では、Layoutが1-Sided PrintingからBooklet Printingに設定が変更されたことにより、CollateがOFFからONへ、GroupがONからOFFへと変化し、Collate、Group、Stapleがdisableし、さらにコンフリクトマークを表示しており、プリンタドライバUIは図8に示すとおりに更新される。処理後ステップ505に戻る。

【0029】ステップ506の判別にて、設定変更要求ではなかった場合、次にグレイアウトの理由表示のイベントであるかを判別する(ステップ512)。本イベントは、例えばコンフリクトマークをクリックした場合やコンフリクトマーク上にマウスカーソルが重なった場合などに発生する。

50 【0030】本イベントが発生した場合、プリンタドラ

イバUI 306はコンフリクトマネージャ303に理由取得の要求を出す(ステップ513)。コンフリクトマネージャ303はコンフリクト処理ルール301を読み込み、指定されたコントロールをグレイアウトしている理由を検索する。

【0031】例えばStapleがグレイアウトされている理由を検索する場合、図6のルールのうちStapleをdisableしているルールを順にチェックする。ここでは(3)

(4)(5)(6)が該当し、(3)のフィニッシャはついておりONのためスルー、(4)のステイブル可能な用紙はOKでありONのためスルー、(5)のBOOKLETは指定されているためこれが理由として該当する、という具合である。なお、ここではフィニッシャが装着されており、用紙はステイブル可能なサイズであるものとした。(6)の用紙タイプがOHPかというチェックは、より優先順位の高い理由が既に見つかっているので実行されない。コンフリクトマネージャ303はこの理由に該当する識別子(IDなど)をプリンタドライバUI 306に返す。プリンタドライバUI 306は理由の識別子に対応する処理を任意に行う。例えば図9に示すようにコメントメッセージを表示することができる(ステップ514)。何かキーを押下したりマウスカーソルを移動するなどによりイベントが解除されると、メッセージは消えステップ505に戻る。

【0032】ステップ512の判別にて理由表示イベントではなかった場合は、続いてプリンタドライバUIのクローズ要求かどうかの判別を行う(ステップ515)。クローズ要求であった場合には、終了処理を行い、プリンタドライバUIをクローズして、全ての処理を終了する(ステップ516)。一方、クローズ要求ではなかった場合には、再び、処理はステップ505に戻る。

【0033】なお、メッセージの表示であるが、カーソルを該当するコントロールの上に移動した際に発生するイベントを利用する。このときに表示するメッセージは、例えば図11に示すようにテーブルとして管理しておく。例えば、コントロールAの場合、そのコントロールAが操作不可になる理由として、図示の場合には2つ存在することを示している。1つは、他のコントロールBがONで、コントロールCがOFFの場合、もう1つはコントロールXがONで、コントロールYがOFFの場合であり、該当する状況に応じて表示すべきメッセージを読み取り、それを表示することになる。

【0034】以上の処理は、プリンタドライバUIがクローズされるまで、繰り返し実行される。プリンタドライバUIがクローズされると処理は全て終了し、本実施形態における印刷処理関連プログラムの処理も終了し、RAM2からはOS205の機能により消去される。なお、本実施形態においては、本印刷処理関連プログラムを記録する媒体を外部メモリとしているが、外部メモリ

としては、FD、HDドライブ、CD-ROMやICメモ리카ード等であってもよい。更に、本印刷プログラム単独、もしくはOSその他のホストコンピュータ上で動作するプログラムと共にROM3に記録しておき、これをメモリマップの一部となすように構成し、直接CPU1で実行することも可能である。

【0035】

【他の実施形態】上記実施形態において、理由表示はステップ514に示したようにプリンタドライバUIが行っているが、図10のステップ1013のようにコンフリクトマネージャ内で行うこともできる。すなわち、コンフリクトマネージャ303の内部に文字列リソースを持ち、該当する理由を検索したらそれをプリンタドライバUI 303に返すのではなく、対応する文字列リソースをコンフリクトマネージャがロードして表示処理を行ってもよい。

【0036】また、ステップ513では最も優先順位の高い理由をひとつ見つけた時点で理由の検索を終了するが、中断せずにルール全体を検索し、該当する理由をすべて表示してもよい。

【0037】以上説明したように本実施形態によれば、コンフリクト処理を行うためのコンフリクト処理ルールを記載するだけで、コンフリクトをかけた処理と同じ条件で理由を取得する事ができる。そのため理由取得のための専用プログラムコードを追加する煩わしさや、条件の漏れや食い違いを生ずるケースから解放され、品質の高い理由表示処理を実現できるという効果がある。

【0038】さらに、理由表示処理をプリンタドライバUIからコンフリクトマネージャに移すことで両者の間で理由識別子の同期を取る必要がなくなり、つまり独立性が向上し、生産性が高くメンテナンス性に優れるという効果がある。

【0039】なお、上記実施形態からの容易に理解できるように、本実施形態におけるその動作は、プリンタドライバ或いはその一部におけるプログラムによって実現できるものである。従って、本発明はコンピュータにプログラムを組み込むことによって実現できるわけであるから、コンピュータプログラムも本発明に含まれる。また、プログラムをコンピュータにインストールする際、フロッピーディスクやCDROM等の記憶媒体を介して行われるわけであるから、それら記憶媒体も本発明の範疇に含まれる。

【0040】

【発明の効果】以上説明したように本発明によれば、所定の処理を行うためのパラメータを決定するために表示されるウインドウにおけるユーザーインタフェースを簡単に構築でき、生産性を高めると共にメンテナンス性にも優れたユーザーインタフェースを提供できる。

【図面の簡単な説明】

【図1】本発明の一実施形態を示す印刷処理システムの

構成を説明するブロック図である。

【図2】ユーザインタフェース制御プログラムを含む印刷関連処理関連プログラムがホストコンピュータ3000のRAM2上にロードされている際のメモリマップを示すブロック図である。

【図3】本実施形態のユーザインタフェース制御部の内部処理の内容を示すシステム概要図である。

【図4】本実施形態におけるコンフリクト処理における各モジュールで扱うデータの関連を示す例である。

【図5】本実施形態におけるコンフリクト処理ルールを使ったコンフリクト処理及びグレイアウト項目の理由表示処理の手順を示すフローチャートである。

【図6】本実施形態における補完処理前のコンフリクト処理ルールの例である。

【図7】本実施形態におけるプリンタドライバの表示するユーザインタフェースの例を示す図である。

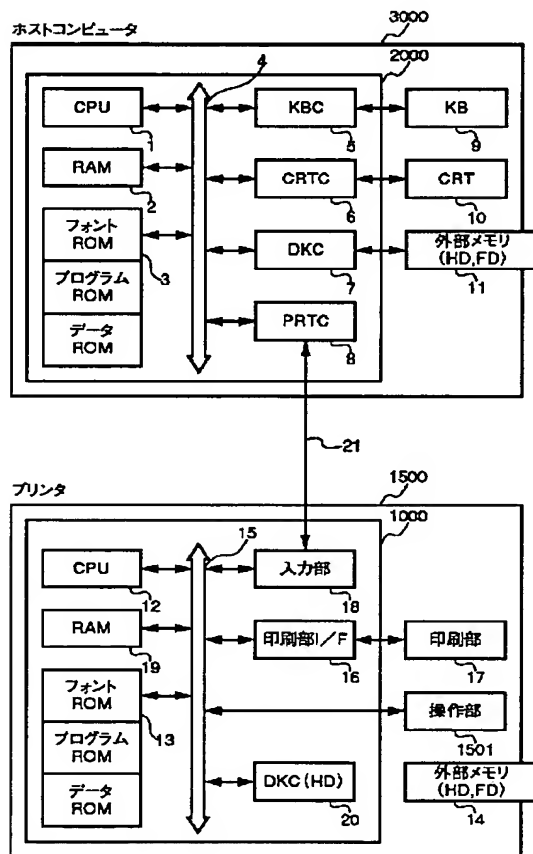
【図8】図7の状態ユーザが[Booklet Printing]を選択した直後の状態を示す図である。

【図9】図8の状態ユーザの操作により理由表示イベントが発生した状態を示す図である。

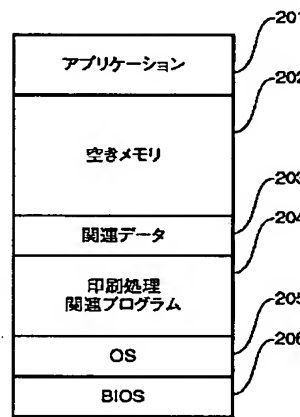
【図10】別の実施形態におけるグレイアウト項目の理由表示をフローチャートにて図示したものである。

【図11】実施形態におけるメッセージルールを示す図である。

【図1】



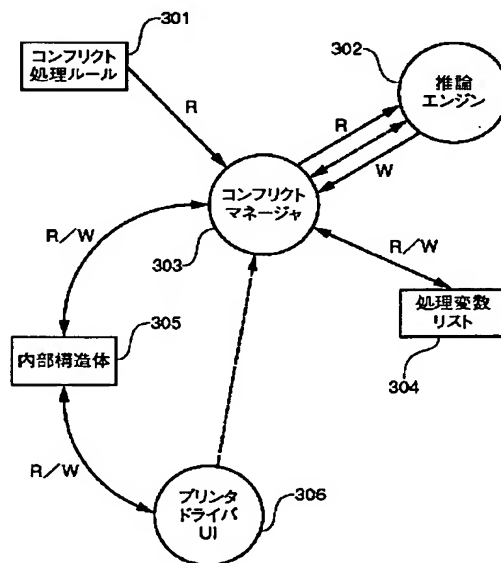
【図2】



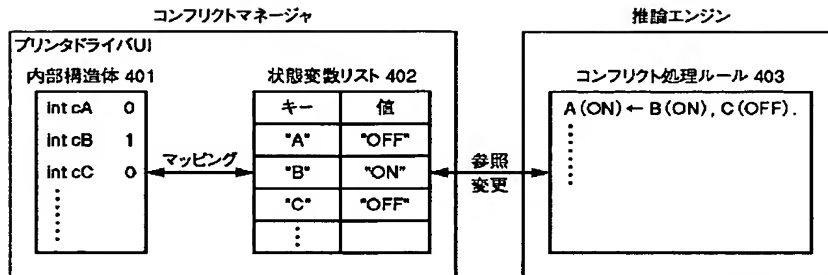
【図6】

- | | | |
|-----|--------------------------------------|-----|
| 201 | Staple(OFF) <- Collate(ON). | (1) |
| 202 | Staple(OFF) <- Group(ON). | (2) |
| 203 | Staple(OFF) <- Finisher(OFF). | (3) |
| 204 | { disable } | |
| 205 | Staple(OFF) <- StapleablePaper(OFF). | (4) |
| 206 | { disable } | |
| 207 | Staple(OFF) <- Layout(BOOKLET). | (5) |
| 208 | { disable } | |
| 209 | Staple(OFF) <- PaperType(OHP). | (6) |
| 210 | { disable } | |
| 211 | Collate(ON) <- Layout(BOOKLET). | (7) |
| 212 | { disable } | |
| 213 | Group(OFF) <- Layout(BOOKLET). | (8) |
| 214 | { disable } | |

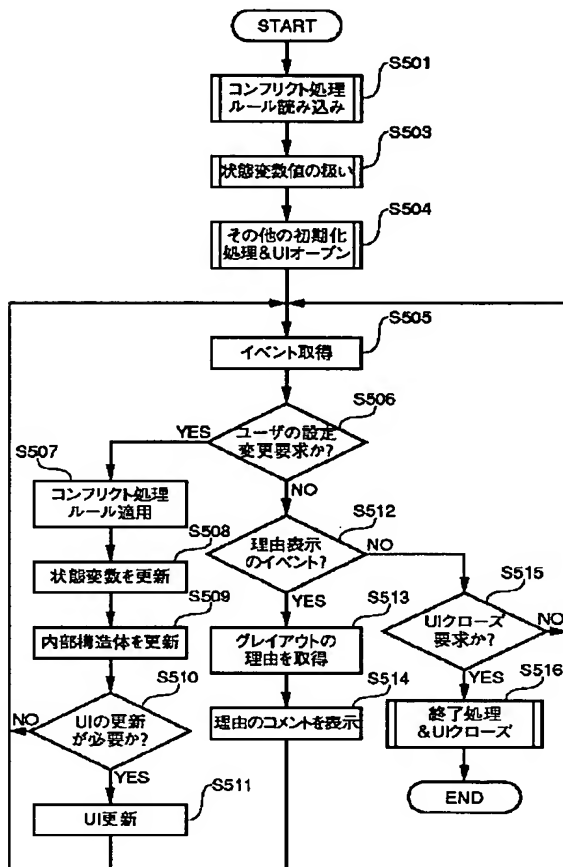
【図3】



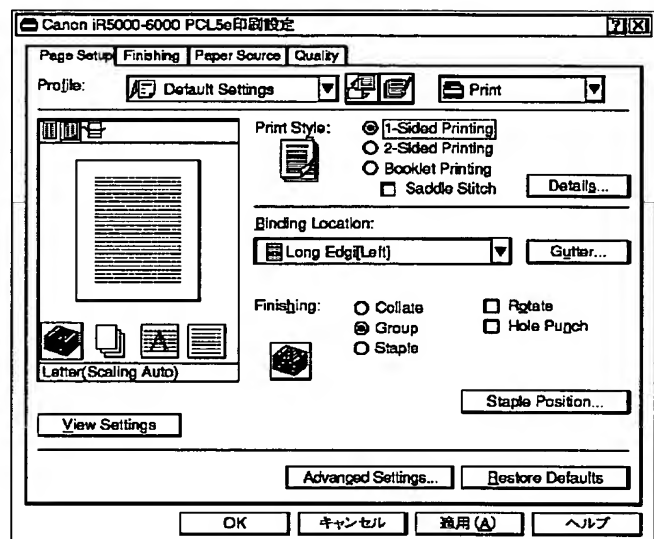
【図 4】



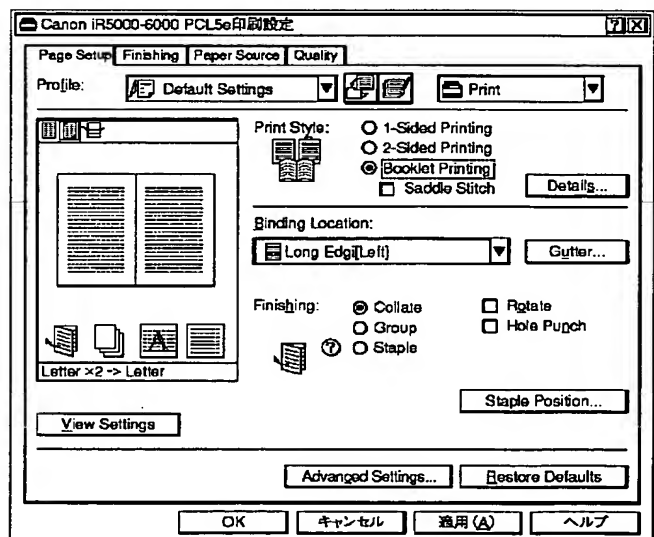
【図 5】



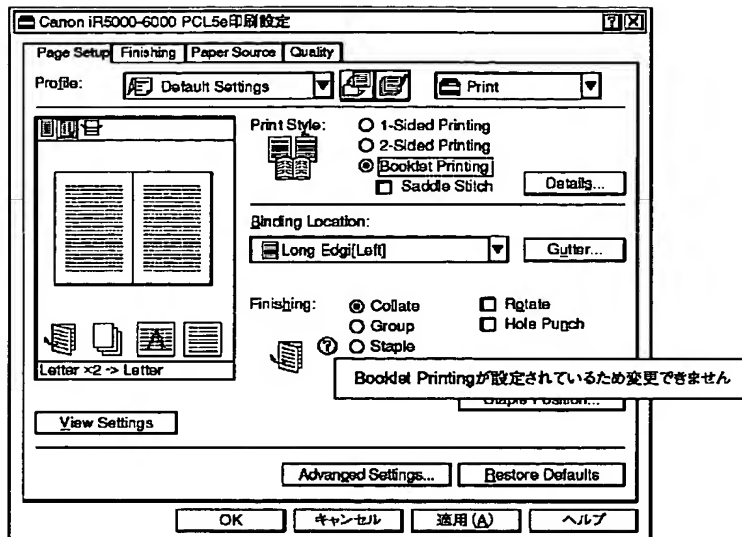
【図 7】



【図 8】



【図 9】



【図 11】

コントロール	disableの条件	メッセージ
...
A	B = NO, C = OFF	XXXが設定されているため、変更できません。
A	X = ON, Y = OFF	YYYが設定されているため、変更できません。
B
...
...

【図 10】

